GPS ロボットカーキット　取扱説明書

1.　はじめに

全地球測位システム（Global Positioning System, GPS）はカーナビや携帯電話に組み込まれるなど、既に身近な宇宙技術の応用となっています。ここでは、GPS の基本的な機能の理解と応用例として、GPS のを使って指定した場所（ウェイポイント）を順番に走行するロボットカーGPS の設計をします。

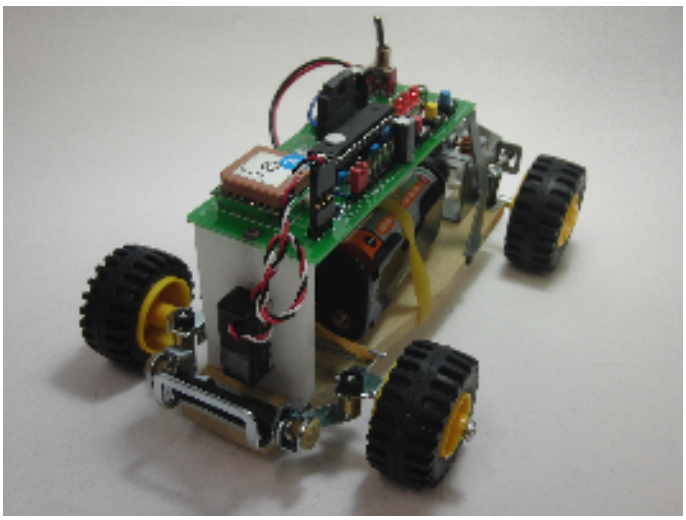

図１　GPS ロボットカーの外観

2.　GPS 受信機について

GPS 衛星は正確な時間を地球に向けて電波ってきます。受信機はいくつかの GPS 衛星から送られてきた時間を読み取るのですが、衛星からの距離が違うとその分遅れて時間を読み取ることになります。その時間差からそれぞれの衛星までの距離を逆算できます。衛星の軌道上の位置があらかじめ分かっていれば、自分の位置が逆算できます。GPS 受信モジュールは図2の写真のものを使います。この GPS は MNEA-0183 データフォーマットと呼ばれる形式でテキストデータを出力する機能を持っています。

図2　GPS 受信モジュールの例(GT-723F)

MNEA フォーマットの中でも RMC（Recommended Minimum Specific GNSS Data）はよく使われるデータが次の例（表1）のようにまとめて出力されています。

$GPRMC,061253.001,A,3252.4844,N,13044.8452,E,008.7,142.0,180110,,,A*6E

表1　GPS の受信例

$GPRMC	ヘッダ
061253.001	測位時刻（UTC） 06:12:53.001
A	ステータス, A:有効, V:無効
3252.4844,N	緯度　32 度 52.4844 分, N:北緯
13044.8452,E	経度　130 度 44.8452 分, E:東経
008.7	対地速度（ノット） 008.7 ノット
142.0	進行方向（度, 真北） 142 度
180110	日付（UTC） 2010 年 01 月 18 日
, ,	地磁気の偏角, 方位 （未測定）
A	モード,　A:単独測位, D:DGPS, N:無効
*6E	チェックサム

GPS ロボットカーでは、この中で緯度、経度、進行方向のデータを抜き出して利用します。目標地点と現在地点との緯度、経緯差から進行方向を決定し、現在のロボットカーの進行方向を合わせるようにステアリング制御することで、目標地点を順にたどる走行が可能となります。

3. 制御回路の説明

キットの回路図と回路基板の部品配置を図３と図４にそれぞれ示します。

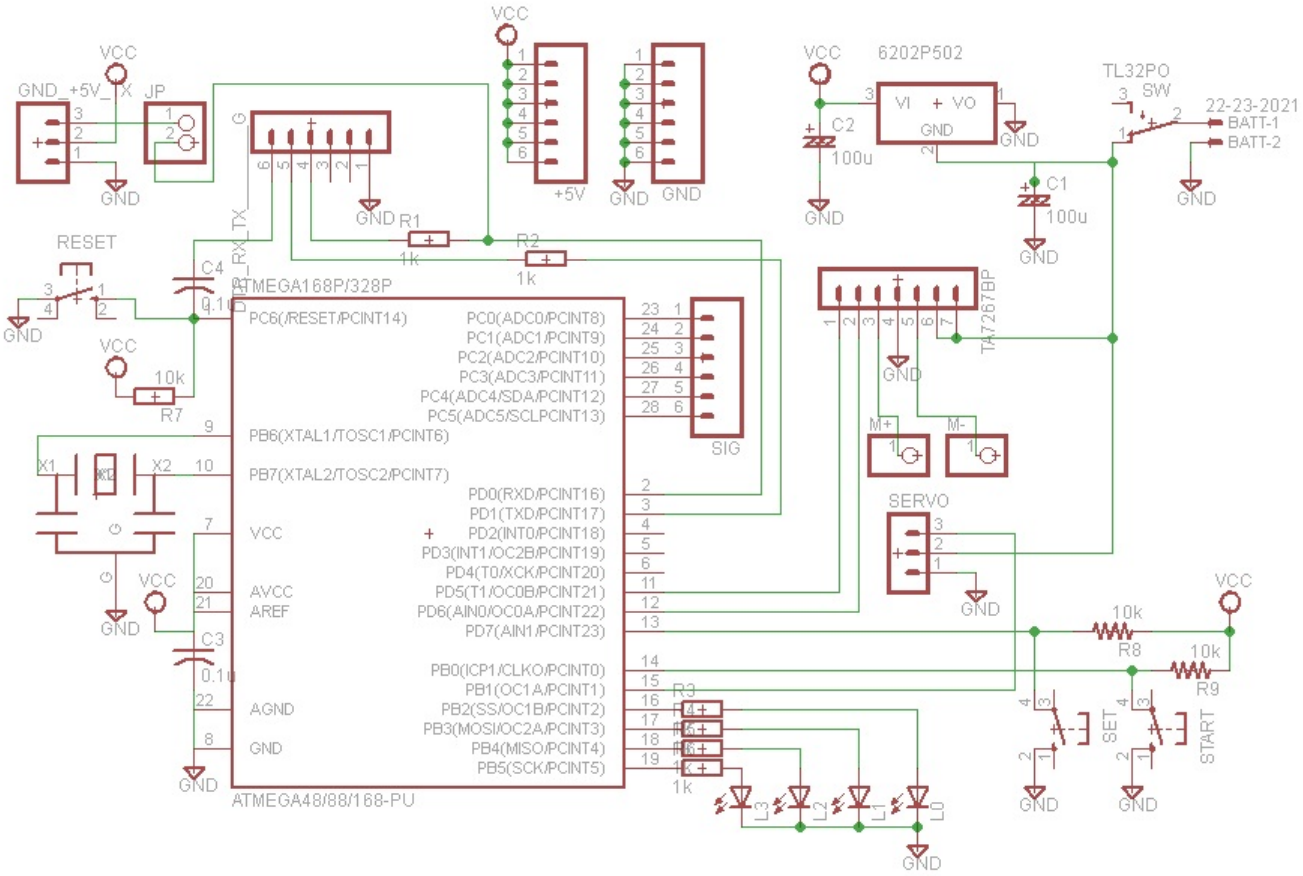

図３　GPS ロボットカーキットの回路

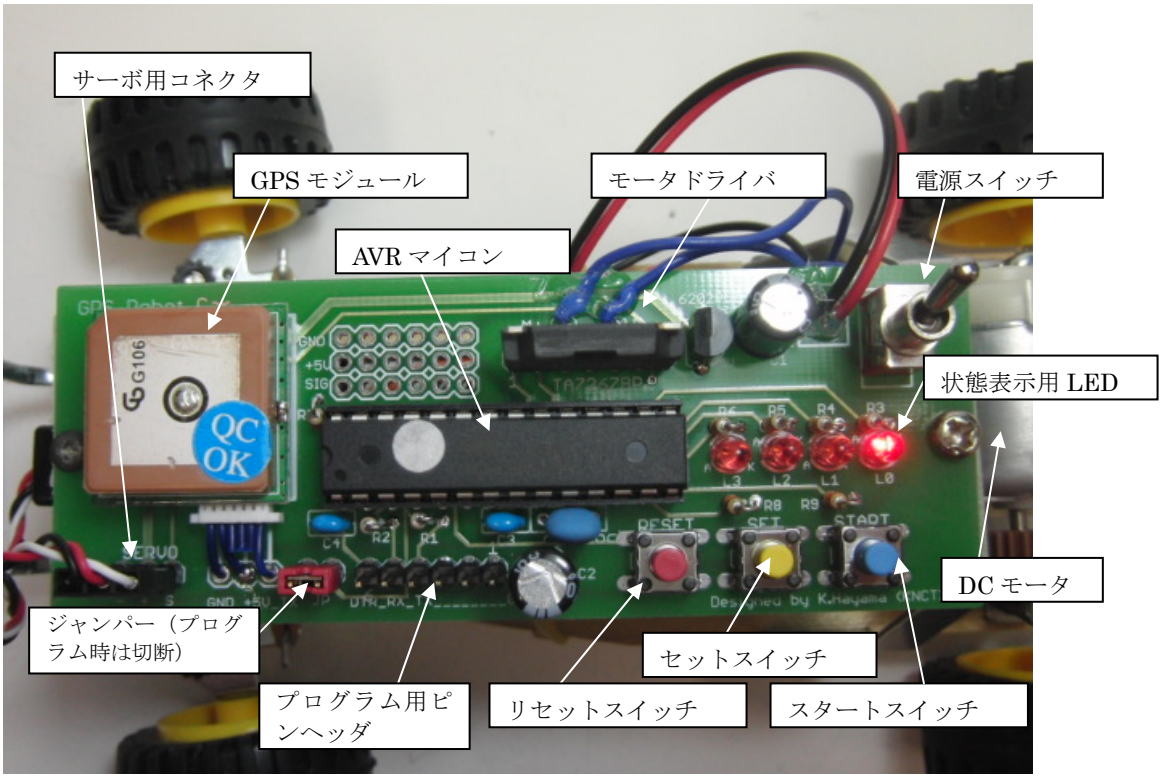

図４　回路基板の外観

制御には AVR マイコン（ATMega328P）を使用しています。単三電池４本を電源に用い、３端子レギュレータより 5V 電源をマイコンに供給します。GPS の信号をマイコンで読み取り、あらかじめ指定されたウェイポイント（WP）にそって走行します。マイコンから出される信号によりステアリング用のサーボとモータの回転を制御します。リセット、セット、スタートの３種類のスイッチが搭載されています（詳しい操作方法は７章で説明します)。制御プログラムはあらかじめマイコンに書きこまれていますが、USB シリアル変換ケーブル（代理店：スイッチサイエンス社 [1]）を使ってプログラムを書き換えることもできます。

（注意：プログラム書き込みの際にはジャンパー線を外して GPS とマイコンを切り離します）

4. 電子基板の製作

使用する部品は表２の通りです。基板の印字に合わせて電子部品を配置し裏面でハンダ付けします。以下の手順で製作してください。

表 2　部品表

	品名	基板上の記号
1	GPS ロボットカー基板	
2	ATmega328P(Arduino ブートローダ書込済)	ATMEGA168/328P
3	GPS モジュール	GPS Receiver
4	カーボン抵抗(炭素皮膜抵抗)1／6W　1kΩ	R1,R2,R3,R4,R5,R6
5	カーボン抵抗(炭素皮膜抵抗)1／6W　10kΩ	R7,R8,R9
6	電解コンデンサー　100 または 220μF　25V	C1, C2
7	積層セラミックコンデンサー　0．1μF　50V	C3,C4
8	3端子レギュレータ	6202P502
9	セラロック(16MHz)	ceralock
10	赤色LED　3mm　OSDR3133A	L0,L1,L2,L3
11	タクトスイッチ	RES, SET, START
12	電源スイッチ	SW
13	ﾋﾟﾝﾍｯﾀﾞ　1x6P	DTR_TX_RX___G
14	ﾋﾟﾝﾍｯﾀﾞ　1x2P	JP
15	ジャンパーピン	
16	IC ソケット(28p)	
17	電池ボックス(単3X4, リード線付き)	
18	バギー工作セット(タミヤ)	
19	アトミックチューンモーター	
20	サーボ	
21	サーボ取り付け板	
22	ピアノ線	
23	ビス, ネジ類	

１）方向に注意してマイコン用の IC ソケット取付けます（基板面の印刷と形状を合わせる)。マイコンは基板が完成してから最後に差し込みます。

２）タクトスイッチ 3 個を 4 端子ともハンダ付けします。3 端子の電源スイッチも取付けます 。

３）抵抗値を確認して抵抗の取り付けます。幅の狭い部分への取り付けは片側の線を 180 度折り曲げ縦に差して取り付けます。抵抗が倒れて他の配線に接触しないように取り付け方向に注意してください。

４）積層セラミックコンデンサ、セラロックを取り付けます。セラロックの中央のピンは GND で左右の極性はありません。

５）シリアル通信用６ピンヘッダ、GPS 用の２ピンヘッダ（JP）を取り付けます。

６）極性のある部品については回路図と基板のパターンの対応を考えながら方向を間違えないようにしてください。

- 電池ボックスのリード線は赤い方を基板の＋（プラス）側、黒い方はー（マイナス）側に取付けます。３端子レギュレータは基板の外側から GND、Vin（電池のプラス側）+5V となります。同等品を使用する場合はピン配置が異なるものがあるので規格表で確認してください。
- 状態表示用の LED4 個は足の長い方がアノード（A）、短いほうがカソード（K）となるように差し込み、裏面ではんだ付けします。
- 電解コンデンサには極性がある（マイナス側が側面に印字されている）ので注意してください。
- モータドライバ（TA7267BP）は印字面を見て左が１番ピンなので、基板と向きを合わせます。
- 図５の写真を参考に、モータを M+と M-端子とに接続します。
- GPS はケーブルを接続して途中でカットし、ケーブル６本中３本を使用します。横から見て左端が GND その隣が＋５V、右端がシリアル出力となります（図６を参照してください）。

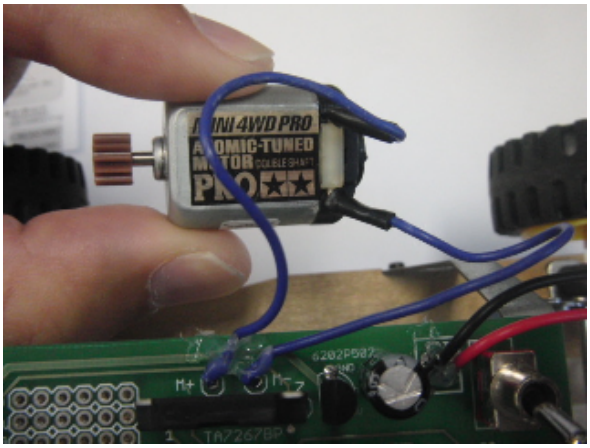

図５　モータと基板との接続

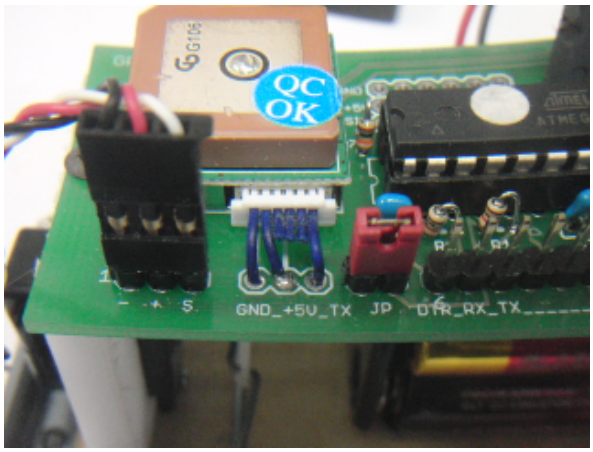

図６　GPS の接続

５．　ベース車体の製作

製作が簡単なバギー工作キット（タミヤ模型）をベース車体としました（図７、図８）。キットは 30 分から１時間位で製作できます。

図７　バギー工作キット

図８　完成したバギー

６．　車体の改造と基板の取付け

マイコンによる自律操縦とするために車体の一部を改造してあります。図９に示すようにステアリングに小型サーボを組込みます。POM 板（白い板の部品）の穴にサーボを固定し、POM 板を立てて車体の下面からネジ止めします。ステアリングの金具の穴とサーボホーンをピアノ線で接続します。また、キット付属のモータでは力が弱いため、ミニ四駆用の強力なモータとの交換してあります。

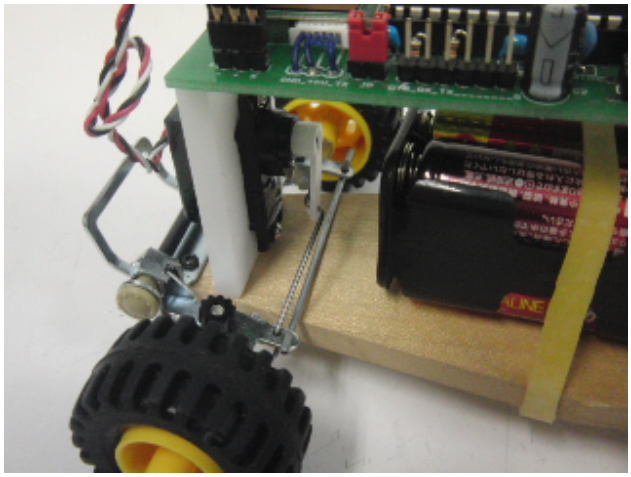

図９　サーボの取り付け

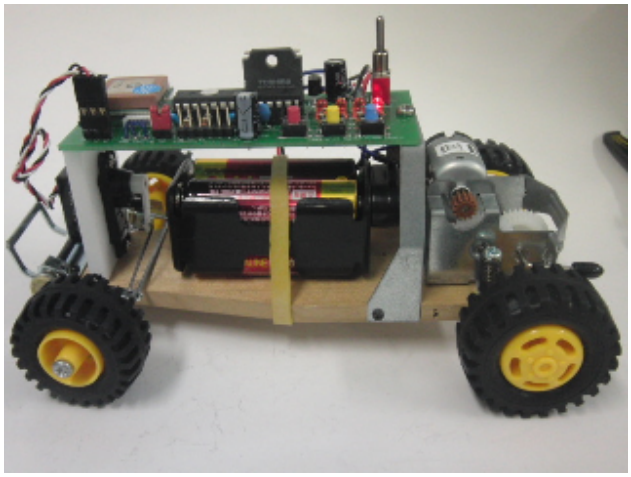

図１０　完成した GPS ロボットカーの外観

図１０は完成した GPS ロボットカーの外観です。基板はバギー後部の金属製のロールバーの上部中央に開けた穴とサーボ固定用の POM 板の上部とで固定します。サーボ用のコネクタを基板に接続したら完成です。

7. 制御ソフトウェアと操作方法

7. 1　制御ソフトウェアの作成

ソフトウェアは Arduino[3,4]の開発環境を用いて作成しました。プログラムを自分で改良したい場合は添付の CD のサンプルプログラムを参照してください。（サンプルプログラムのままで動かす場合はこの章の説明は飛ばしてかまいません）

最初に必要なヘッダファイルをインクルードした後、ピン番号の定義、そのほか各種定数を定義しています。走行の具合に応じて、WP の半径、ステアリング舵角の係数、走行スピード（モータの PWM 値）などを調整するとよいでしょう。

Setup()内で初期設定を行い、loop()内でプログラムモードの選択と実行プログラムの呼び出しを行なっています。loop()以降には、GPS 受信、モータの正転・逆転、サーボのテストプログラム等があります。主要な関数を以下に説明します。

recvGPS()関数は GPS からデータをシリアル通信により受信します。受信バッファを調べ、空でない場合は recvStr()関数により改行コードを受信するまで文字列を読み取ります。読み取った文字列の最初に＄GPRMC を見つけると、文字列の中から緯度、経度、方位、速度を文字列として抜き出します。その後、gps_val()関数により文字列から数値化を行い、続いて gps_cal()関数により現在地と WP との関係を計算します。

WP の設定は setPoint()関数により行います。現在地の緯度・経度を倍精度実数で読み取り、配列に記憶すると同時にバイト単位に直して EEPROM に記憶します。倍精度実数の読み書きには、Arduino フォーラム上のサンプル[3]を利用しました。EEPROM_writeDouble()関数によりデータの記録、

EEPROM_readDouble()関数によりデータの読み出しを行っています。

GPS による走行は runGPS(引数)により行います。最初のウェイポイントを読み出し、モータを正転させてロボットカーを前進させます。recvGPS()関数を呼び出し GPS が受信できた時には、目標地点までの距離の目安を LED に表示させ、角度差に応じて方向にステアリングを切ります。続いて WP までの距離を調べます。設定された半径内に入った時はロボットカーを一旦停止させ、次の WP に更新した後に再スタートさせます。ステアリングの制御は GPS の受信毎に 1 秒間隔で行います。すべての WP を通過したらロボットカーを停止させて制御を loop()関数に戻しています。

そのほか詳細な内容は直接プログラム及びコメントを参照してください。

7.2 操作方法

基板上には、リセットスイッチ、セットスイッチ、実行スイッチの 3 種類のスイッチがあります。初めて動作させる時はセットスイッチを押しながら電源を投入してください。この操作でパラメータがすべて初期値に戻ります。セットスイッチでプログラムモードを選択すると、状態表示用の LED にモードが 2 進数で点灯表示されます。スタートスイッチで選択されたモードを実行します。動作を中断させ初期状態に戻るにはリセットを押します。用意したプログラムモードとその説明を以下の通りです。

表３　プログラムモードと動作

0	サーボテスト サーボが右と左に各 1 秒ずつ振れてニュートラルに戻る	8	ステアリングの最小値の設定 スタート、セット SW で増減。リセットで戻る
1	モーターテスト 実行 SW でモータ正転、セットスイッチでモータ逆転	9	ステアリングの最大値の設定 スタート、セット SW で増減。リセットで戻る
2	GPS テスト GPS の信号を受けて LED が 1 秒おきに点滅 １個点滅：GPS 捕捉なし、２個：捕捉済み	10	WP の半径設定 スタート、セット SW で増減。リセットで戻る
3	WP の設定 セット SW でその場所を WP として追加（最大 16 箇所）。EEPROM にも記録。	11	舵角計算の比例値設定 スタート、セット SW で増減。リセットで戻る
4	WP のクリア	12	走行スピードの設定 スタート、セット SW で増減。リセットで戻る
5	GPS 走行(方位がとれなくても偏差で計算)	13	WP のシリアル出力 GPS のジャンパー外してシリアル接続
6	GPS 走行(方位がとれなかった時はニュートラルに戻す)	14	パラメータのシリアル出力 GPS のジャンパー外してシリアル接続
7	ステアリングニュートラル位置の設定 スタート、セット SW で増減。リセットで戻る	15	予備

参考文献

[1] http://www.switch-science.com/

[2] http://www.arduino.cc/

[3] http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1234477290/3

キット設計　葉山清輝（熊本高専）

質問等ありましたら、hayama@kumamoto-nct.ac.jp までメールください。